REMARKS

Claims 1-50 are pending in this application, all of which have been rejected. In view of the following remarks, Applicants respectfully request reconsideration of the application.

Response to Arguments

In paragraph 1, the Examiner states that the arguments provided in the response filed April 6, 2006 have been considered but are not persuasive. Applicants disagree with the Examiner's assessment of the arguments, and will address them in detail below.

Rejections under 35 U.S.C. §102

In paragraph 4, the Examiner rejected claims 1-50 under 35 U.S.C. §102(e) as being anticipated by *England* (USPN 6,775,779). Applicants traverse.

Claims 1-10 and claims 24-30

As per independent claim 1, the Examiner finds all elements to be taught by *England* et al., and cite col. 2, line 66 through col. 3, line 13 and col. 9, line 55 through col. 10, line 5 for support.

England does not teach determining if a file is secured

Claim 1 recites in part "determining…whether ***the file being accessed is secured.***" (emphasis added). In exemplary embodiments, secured files are defined as a "type of digital asset that cannot be accessed without a priori knowledge. Example of a priori knowledge may include, but not be limited to, a password, a secret phrase, biometric information or one or more keys." (see [0026]) Furthermore, the secured files, in exemplary embodiments, are files that may be partly encrypted and may contain access rules which define access privileges. Embodiments of the present invention determine if files, not applications or modules that operate on the files, are secured.

In contrast, the cited portions of *England* refer to "running designated processes, libraries, or other software components at a higher level of protection" (col. 2, line 64 – col. 3, line 1). *England* describes as examples, "rights-management operation-system modules, communication drivers, and video decoding applications programs" which can run in protected memory that is not accessible by other modules (col. 3, line 3-6). Thus, *England* is concerned with protecting applications, "where the application must be protected from viruses or other malicious code" (col. 3, line 26-27).

Even if the Examiner is to argue that the applications are files, these ***applications are not secured***. If these applications were secured, there would be no need to protect these applications by running them in a protected memory.

England does discuss having trusted modules that exchange data among themselves, whereby trusted modules can identify other trusted modules (col. 3, line 9-17). These modules, however, are defined as "packages of executable instructions and data which may perform functions for OSS or for applications" (col. 5, line 25-26). Furthermore, these ***modules are not secured***; they are only trusted. Thus, *England* is directed to determining trusted modules and protected applications, not determining if files are secured.

England does not contemplate determining whether the file being accessed is secured. The equivalent "files" in *England* are the content, which are defined as "digital data such as movies, music, and other media presentations that third parties make available on media or by download for us in computer" (col. 5, line 29-31). However, there is no distinction in *England* between secured and non-secured content. Because

England does not contemplate any distinction between secured files, applications, or modules, it cannot teach determining if files are secured.

England does not teach differential treatment of files

Claim 1 further recites "when the file is determined to be secured, activating a cipher module and loading the file through the cipher module into the application; and when the file is determined to be non-secured, loading the file into the application without activating the cipher module." In exemplary embodiments, secured files are defined as a "type of digital asset that cannot be accessed without a priori knowledge. Example of a priori knowledge may include, but not be limited to, a password, a secret phrase, biometric information or one or more keys." (see [0026]) If the files are secured, then a cipher module is utilized to operate on the secured file. For example, a file key may be retrieved from the security information and used to decrypt the encrypted data portion in the selected secured document by the cipher module. (see [0098]). As such, the cipher module is configured, in various embodiments, to generate a file key, encrypt content in a document/file with the file key, and/or decrypt a secured file. However, if the file is not secured, then the cipher module is not utilized.

England does not contemplate differential treatment of the file (i.e., content) based on whether the file is secured or not. That is, all files in *England* are treated the same way via the trusted module/application architecture described in *England*.

Because, *England* does not contemplate differential treatment of the file based on the file being secured or not, claim 1 is not anticipated by *England*. Further, claims 2-10 which depend from claim 1 are not anticipated by England for the same reasons as claim 1.

Claim 24 is rejected by the Examiner for the same reasons as claim 1. Claim 24 recites in part "when the file is determined to be secured, activating a cipher module that operates in the operating system; loading the file through the cipher module into the application." As discussed above, *England* does not contemplate determining if a file is secured or the differential treatment of the secured files. Therefore, claim 24 is not anticipated by *England*. Additionally, claims 25-30, which depend from claim 24, are not anticipated for the same reasons as claim 24.

<u>*Response to Arguments*</u>

The Examiner points to col. 3, lines 14-17 and 35-42 and col. 9, lines 30-39 for support for determining whether the file is secured. However, col. 3, lines 14-17 merely discusses the trusted module architecture whereby one trust module selects other modules which it trusts. This does not support or suggest any teaching of a secured file or even determining if a file is secured.

Col. 3, lines 35-42 merely discusses the prevention of tampering with code (i.e., changing the code) in such a manner that a modified program will be unable to authenticate itself as trusted. Once again, this does not teach or suggest a secured file, as taught by embodiments of the present invention, or even determining if a file is secured.

Finally, col. 9, lines 30-39 merely teaches the trusted environment whereby a trusted application may select other trusted components. As discussed above, the trusted module/application environment of *England* does not contemplate distinguishing between secured and non-secured files or differential treatment of the file based on the determination of whether the file is secured or not.

<u>Claims 11-19</u>

With regard to independent claim 11, the Examiner cites to the same section of *England* for support as that of claim 1. Claim 11 recites in part "maintaining a file key in a temporary memory space" and "preparing security information for the encrypted portion, the security information being encrypted and including the file key and access rules to control access to the encrypted portion" of the file.

As previously, discussed, the cited portions of *England* (along with the rest of the patent) refer to protecting applications and modules from viruses and other malicious code. The cited portions of England do not discuss maintaining a file key in a temporary memory space, nor do the cited portions teach security information being encrypted and including the file key and access rules to control access to the encrypted portion of the file.

Therefore, claim 11 is not anticipated by *England*. Claims 12-19, which depend from claim 11, are also not anticipated by *England* for the same reasons as for claim 11.

Response to Arguments

In maintaining the rejection, the Examiner cites to the same portions as referenced for the rejection of claim 1 and to col. 12, lines 40-57 and col. 14, lines 10-17. However, col. 12, lines 40-57 discusses making programming of "the access-control table possible only when secure loader 410 is executing." The programming of the access-control table is not equivalent to "maintaining a file key in a temporary memory space" or "preparing security information for the encrypted portion, the security information being encrypted and including the file key and access rules to control access to the encrypted portion.

Col. 14, lines 10-17 discusses a set of permissions for different CC-register contents stored in a page of memory. Thus, these permission sets are associated with memory pages of a memory, not security information that is encrypted within a file.

Claims 20-23

With respect to independent claim 20, the Examiner once again cites to the same section of *England* for support as that of claim 1. Claim 20, however, recites in part "determin[ing] whether the file being accessed is secured" and "activating a cipher module when the file is determined to be secured."

As previously discussed, *England* does not contemplate differentiating a secured file from a non-secured file. As such, *England* cannot contemplate different treatment of the file based on whether the file is secured or not (i.e., activating a cipher module when the file is secured). Therefore, *England* cannot disclose determining whether the file being accessed is secured or activating a cipher module when the file is determined to be secured. As such, independent claim 20 is not anticipated by *England*. Additionally, claims 21-23 are also not anticipated by *England* by way of their dependency from claim 20.

Claims 31-39

With respect to independent claim 31, the Examiner also cites to the same section of *England* for support as that of claim 1. Independent claim 31, however, recites in part, "encrypting the file with the file key in a cipher module to produce an encrypted file."

The cited portions of *England* do not provide any support for a cipher module. For example, cited col. 9, line 55 through col. 10, line 5 teaches an interrupt handler, not a cipher module. The interrupt handler "saves the system state in a secure page, so that no other code can access it or any secrets it might contain." The handler "can then initiate a software interrupt that the untrusted part of the OS is allowed to process." Clearly, the handler is not a cipher module and does not perform the functions of the cipher module. Therefore, independent claim 31 is not anticipated by *England*. Further, claims 32-39, which depend from claim 31, are not anticipated by *England* for the same reasons as that of claim 31.

Claims 40-50

In regards to independent claim 40, the Examiner cited to the previously cited portions of *England* along with col. 6, line 33-45. As previously discussed, col. 2, line 66 through col. 3, line 13 and col. 9, line 55 through col. 10, line 5 of *England* refer to the protection and utilization of applications and modules. Col. 6, line 33-45 merely teaches an access-control table which contains bits that determine rights for a program combination that accesses a page. For example, one bit of each entry indicates whether the programs specified have read, write, or execution privileges for the page specified. Once again, the cited portion only refers to program protection and access.

Claim 40 recites, in part, "a cipher module activating upon determining that the file being accessed is secured." As discussed with respect to claim 1, England does not contemplate determining if a file is secured and further does not contemplate treating a secure document different from an unsecured document. As such, England cannot contemplate activating a cipher module upon ***determining that the file being access is secured***. *England* does not anticipate claim 40. Further, claims 41-50, which depend from claim 40 are not anticipated by England for the same reasons as that of claim 40.

Conclusion

Based on the above amendments and remarks, Applicants believe that the rejections in the final Office Action of June 29, 2006 are fully overcome, and that the application is in condition for allowance. If the Examiner has questions regarding the case, the Examiner is invited to contact Applicant's undersigned representative at the number given below.

Respectfully submitted,

Chang-Ping Lee et al.

Date: 9\29\06

By: Susan Yee

Susan Yee, Reg. No. 41,388
Carr & Ferrell *LLP*
2200 Geng Road
Palo Alto, CA 94303
TEL: (650) 812-3465
FAX: (650) 812-3444